AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning on page 16, line 7, with the following:

Five types of user interface units 201 are shown in FIG. 2. The primary user interface unit is, preferably, a wall mounted unit 132, 162 that is hardwired to a trunk, such as 170. These may come in several sizes, such as a large size 132 and a smaller size 162. They may also come in a tabletop unit, which includes a tripod to hold it upright and an extension cord that plugs into a wall jack. Another type of user interface unit 201 is a variable output device, which we refer to herein as a level control 140, or simply a level control. Often, a power level control 140 is referred to in the art as a "dimmer", since, when used to control lights, they can dim the lights. However, herein, level controls 140 may also control devices other than lights, such as a fan, or other aspects of the system, such as an audio level, or the temperature of a room. Generally, a level control controls the power to a device, though it may also control another electrical parameter. A further type of interface unit [[20]] 201 is a wireless remote control 150, which preferably communicates with a level control 140 or a wall mounted interface unit 162 via infrared (IR) radiation 151. User interface unit 150 is preferably small, most preferably about the size of a key fob. Yet another type of user interface unit 201 is a larger wireless remote unit 176, which communicates via radio frequency (RF) radiation 178 with a transceiver 177 that is hardwired to a trunk, such as 159. Interface unit 176 is preferably about the same size as the smaller wall-mounted interface units 162. The smaller remote may be an RF remote, and the large an IR remote, or both the smaller unit 150 and the larger unit 176 may use both IR or RF. Preferably, all the interface units 201 are in some way connected to a trunk line, such as 170 or 159.

Please replace the paragraph beginning on page 18, line 1, with the following:

Turning now to the details of FIG. 3, building automation system 120 includes a controller 122 connected to a breaker panel 121, via AC wiring 125. Controller 122 is preferably housed in a cabinet 172 which preferably includes at least one module bay 173, as shown in FIG. 6, for holding additional low voltage electronic driver modules 117.

Breaker panel 121 provides electrical power to drivers 127, 143, 158, 206, 220, <u>222, 229,</u> 231, [[222,]] and 237 via AC wiring 128, 124, 123, 207, 221, 230, 223, 228, and 236, respectively. Power driver 127 provides controlled electrical power to lighting fixtures 129, 139, 142, and 166 via AC wiring 130, 138, 134, and 135 respectively. Power driver 143 supplies controlled electrical power to lighting fixtures 145 and 165, light 148, fan 149, recessed ceiling lights 167, via AC wiring 144, 146, 147, and 141, respectively. Power driver 158 supplies controlled electrical power to fireplace control 154 of fireplace 28, lighting fixture 157, lighting fixture 163, switched outlet 164, via AC wiring 155, 156, 160, and 161, respectively. Power driver 206 supplies controlled electrical power to shades 208 which control solar heating, windoware 209, air conditioner 210, and intercom/speaker/microphone system 211 via the indicated AC wiring. Power driver 220 provides controlled electrical power to furnace 212, air control dampers 213, hot water heater 214, and aquarium heater 215 via the indicated AC wiring. Power driver 222 provides controlled electrical power to garage doors 216, electric gate 217, heated sidewalks and driveway [[219]] <u>218</u>, and pool/spa heaters 219 via the indicated AC wiring. Power driver 229 provides controlled electrical power to sprinklers 224, drapes 225, door locks 226, and digital camera 227 via the indicated AC wiring. Power driver 231 provides controlled electrical power to fire detection system 232, flood monitoring system 233, humidity controller 234, and outdoor lighting 235 via the indicated AC wiring. Power driver 237 provides controlled electrical power to home theater audio/visual components 238, phone PBX 239, voice mail control 240, and doorbell 241 via the indicated AC wiring. Similarly, it is evident that any electronic or electrical device in a building can be driven by an appropriate driver.

Please replace the paragraph beginning on page 18, line 30, with the following:

Drivers 127, 143, 158, 206, 220, 222, 229, 231, and 237 receive the data needed to set the power level provided to the driven devices from controller 122. The data is provided to the controller by a network of user interface units 132, 140, [[150]] <u>152</u>, 162, 168, 169, and 176 (via transceiver 177) via low voltage control trunks 159 and 170.

Specifically, user interface units 132, 140, 152, and 162 are connected with controller 122 via low voltage control trunk 170. User interface units 168, 169, and transceiver 177 are connected with controller 122 via low voltage trunk 159. In FIG. 3, the RF wireless remote control 176 is shown located in family room 44, but it can be moved throughout the building. It communicates with transceiver 177 located on patio 47. In the simplified building shown in the drawing, only one transceiver 177 is necessary. Generally, there will be several transceivers 177 placed strategically about a building so that RF wireless interface 176 anywhere in or about the building can communicate to controller 122. Also preferably, building automation system 120 includes IR wireless remote control 150 that also can be moved about a building from room to room. IR wireless remote control 150 communicates with controller 122 via user interface unit 162 located in the room in which remote control 150 is used. In FIG. 3, wireless remote control 150 is shown located in bedroom 45.

Please replace the paragraph beginning on page 19, line 29, with the following:

FIG. 4 is an illustration of the preferred embodiment of controller 122. Controller 122 preferably comprises an electronic processor 242, bus 243, memory connector 244, memory 245, RS-232 port 249, a small battery (not shown) to provide power to the clock, and trunk drivers 260 and 261. It also includes a connection 246 to power supply 174 and backup battery 175 in cabinet 172 (FIG. 6). Preferably, memory connector 244 includes a plug/socket 248 for connecting to a memory, such as a Flash memory. Preferably, memory 245, memory connector 244, and trunk drives 260 and 261 are connected to processor 242 via bus 243. The power supply is connected to processor 242 and to memory 245, memory connector 244 and trunk drives 260 and 261, though the latter connections are not expressly shown in FIG. 4 for clarity. In the preferred embodiment, processor 242 includes a microprocessor and a digital signal processor. The microprocessor is preferably a Microchip model PIC 185452 with embedded EEPROM and the digital signal processor is preferably a TMS 320VC5502 digital signal processor. Memory 245 preferably includes a SDRAM and a NAND FLASH memory. In the preferred

embodiment, the SDRAM is a Hitachi model HM5257165 and the NAND Flash memory is a Toshiba TH58100FT. Memory connector 244 is preferably an MMC card connector such as commonly used in electronic cameras and music players. Plug/socket 248 may be either a plug adapted to connect to a socket or a socket adapted to connect to a plug. The important aspect of plug/socket 248 is that it enables quick and simple connection of a removable non-volatile memory medium, such as will be no problem for someone not skilled in computers to use. Non-volatile medium 533 (FIG. 5) may be any non-volatile memory medium, such as a hard drive, a tape, a disk, a card, etc. Trunk drives 260 and 261 are MAX drivers. They communicate at either a 250 K-baud or 500 K-baud rate, whichever is selected, with full cyclic redundancy check error checking. Power supply 246 provides 24 V DC power from a conventional 110-120 V AC input, and battery [[247]] <u>175</u> provides 24 V DC backup power and is rechargeable. While only two trunks are expressly shown in FIG. 4, dots 259 represent four additional trunk drivers and trunk lines in the preferred embodiment for a total of six. Each trunk preferably is capable of driving 50 devices. The invention contemplates a slave controller (not shown), which can be included in building automation system 120, particularly in large buildings. In this case, the slave controller would be located on one trunk, and have a plurality of additional trunks attached to it. However, the building automation system is preferably designed for one controller 122.

Please replace the paragraph beginning on page 24, line 25, with the following:

User interface unit 162 is low voltage and, as discussed in connection with FIG. 4, is system powered via trunk line 170. It preferably includes a UL-rated circuit board that preferably incorporates a 16-bit, 16K microprocessor 1020, preferably with an embedded Flash memory 1022. In the preferred embodiment, it includes a PLCC-44 microprocessor with embedded EEPROM and a Microchip model PIC 18F452-I/L in the basic unit. An enhanced unit also includes a Toshiba model TC58FVB160AF chip with an embedded 1MX16 EEPROM Flash memory and a Toshiba model TH58100FT chip with a 128 X 8 NAND EEPROM 48-TS. A user interface may also include a TMS 320VC5502 digital signal processor. The electronics of each user interface unit 162 preferably also includes

software, which is object based and table driven. This software is used to program and display the display 250. Housing [[281]] 251 is preferably about 4.6 inches wide by 5.7 inches high and 0.75 inches thick.

Please replace the paragraph beginning on page 29, line 12, with the following:

As indicated above, when system 120 is installed, every slave device 201, 202 is connected to a trunk. When the system is commissioned each slave device, including all user interfaces 201 and all drivers 202, including all add-on cards 117, is assigned an address. Communications between controller 122 and slave devices 201, 202 follow a novel protocol, the data packet 420 that is illustrated in FIG. 20. Periodically, preferably every 50 milliseconds (msec), controller 122 communicates a global synchronization signal 422. Global synchronization signal 422 includes an indication that a new message sequence is starting, time and date data, and may include other global data. Accurate time and date data is maintained within controller 122. Data may optionally be obtained from WWV receiver 118 (FIG. 2). Following the synchronization signal, data packets 424, 432, 440, 444, etc., are sent sequentially. Each data packet includes an address and a data block; i.e., data packets 424, 432, 440, 444 include addresses 426, 434, 441, and [[456]] 446, and data blocks 428, 436, 442, and 448, respectively. Additional data packets are sent until all electronic elements in the system have been addressed, as indicated by dotted line 449. After each data packet is sent, controller 122 waits for a short period 430, 438, 447, etc., for the addressed electronic system element to respond. The response is also in packet form, including the controller address and data confirming that the addressed element is responding, and providing information requested by the controller. If a message sent by the controller or an electronic element is too long for a single data packet, it is broken up into as many data packets as are needed to complete the message. For example, when new software is uploaded from the controller to user interfaces 201 or drivers 202, many data packets are required.

Please replace the paragraph beginning on page 31, line 3, with the following:

Turning now to the details of the software and the programming of system 120, as indicated above, when a new system is first installed, the first part of the programming process is a task called "commissioning". The commissioning process is shown in FIG. 21. During this process, the various slave devices in the system will be identified and given names that identify their physical location within the system. For a controller to exchange information with specific slave devices, that slave device must have a unique "address" of some kind. As indicated above, this address is used as part of the message to the device to indicate that the data is intended for or requested from that device and that device only. As also indicated above, each slave 201, 202 and controller 122 contains an EEPROM memory device either as a discrete component or as part of a microprocessor chip. Several locations of the EEPROM memory are reserved for the "commissioning date" and several more are reserved for the "device address". These are left blank at the time of production. The slave devices are installed and connected to a trunk 170, 159 at 452. Initially, no slaves respond to any controller messages. At 454, the installer or user touches the commissioning "button" on a touch screen, which preferably is any part of the touchscreen. Each slave includes a "commissioning button". On touchscreens 132, 162, 176, pressing the quick-access button or the touchscreen serves this purpose. On level controls 140, pressing either position of the rocker serves. Drivers 202 and other devices without other buttons have a button specifically dedicated to commissioning. The first touchscreen touched becomes the default programming terminal for the system. It is first used in process 456 to set the time and date values of the master clock in the controller. The date so set becomes the commissioning date. The commissioning date is repeatedly broadcast to all slaves 201, 202 as part of the global sync message sequence 422 and the slaves store this date in local memory. Further, the pressing of the commissioning button sends a Requested Data Returned to the controller from the slave device. At 458, the controller sends a screen to the programming touchscreen that requests data for the device, including room and wall locations. The installer or user enters this data at 460, and commissioning of the device is initiated. In an address set process 462, the [[The]] controller sends a Send Data message to address zero with the byte 2 parameter of the

message set to 4. (see below). The data of this message is the "working address" to which the device will be set and referred to for all future communications. The device saves that address to EEPROM along with the commissioning date. The controller determines if the device is a touchscreen device at 464. If the device is not a touchscreen user interface, the controller sends the screen requesting data for the device to the programming touchscreen, the data is entered on that touchscreen, and the address is set at the device to be commissioned. If the device is a touchscreen, the data request is displayed 474 on the new touchscreen device to be commissioned, and the address is set as before. The installer or user then touches the commissioning button of the next device at 464. This continues via processes 466 and 464 until no further commissioning button is pushed, and the commissioning process ends. In this manner, the controller assigns a unique working address to each device. Anytime a new device is added to the system and its commissioning button is pressed, the commissioning screen appears automatically. At this time, all the devices are given a new commissioning date.

Please replace the paragraph beginning on page 32, line 25, with the following:

Once each slave device has been assigned an address, the system can be programmed. The commissioning process places a list of all devices and where they are located in controller 122. A programming button is also placed on each touchscreen. The programming process is shown in FIG. 17. First, program key 1110 of one of interface units 132, 162, 176 is touched 302. The controller sends a display list 195 (FIG. 15) to the interface unit, the program key of which was touched, and this list is displayed 304. The user touches 306 the device on the display that it is desired to program. Controller 120 determines if this is a touchscreen device at 308, and if it is, the user is then, at 310, allowed to add or delete screen objects to be programmed. This is done by displaying the available objects, such as level control switches or controlled devices, and using buttons at the bottom of the touchscreen to indicate whether the objects should be added or deleted. When the touchscreen is entirely programmed, the software automatically arranges 312 buttons for the devices and objects on the touchscreen. The buttons are automatically

arranged in the order in which they were added, displaying each in the next available space from left to right and then from top to bottom. Buttons that need to be further programmed are touched at 314, and the system displays a list of attributes at 316. These attributes include what interface or interfaces 201 are intended to control which drivers 202, or conversely, which drivers are to respond to which interfaces, the default settings of level controls, the default settings of other electronics, minimum and maximum values, times at which functions may be turned on or off, etc. The list of attributes, while shown following the automatic arranging of buttons, can also be accessed prior to the arranging of buttons, and in fact at any time in the programming after commissioning provides a list of devices. Usually, programming is an iterative process in which objects and attributes are selected and programmed, rearranged, deleted, reselected, etc., until the screens are exactly as the user desires. The attributes are programmed at 318. If the device is not a touchscreen, the program proceeds immediately to displaying 316 the list of attributes. When the selected device is programmed, list 195 is again displayed, and the user can elect to program another device at 320, by touching 306 a device on the display, or exit from programming mode and return 322 to normal mode.

Please replace the paragraph beginning on page 40, line 7, with the following:

As indicated above, the protective codes are preferably included by a user interface [[101]] 201 and device driver [[102]] 202 (FIG. 2) in the Requested Data Returned message during the commissioning process. Controller 122 uses this code to indicate what licensed manufacturer or distributor made each device in the system, and to provide a file of manufacturers and licensees the devices of which are included in the system. If manufacturer or distributor is not the owner of the technology or a licensee, the controller will not recognize nor communicate with the device and it will not operate within the system. An alarm or other indication that a non-qualified device has been connected to a trunk may also be output.

Please replace the paragraph beginning on page 42, line 6, with the following:

The controller firmware also includes a novel process <u>350</u> for copying a program, updating firmware for all devices, or backing up system 120. The process <u>350</u> for copying a program or updating system firmware is shown in FIG. 18, and the process <u>400</u> for backing up the system is shown in FIG. 19. First, at 352 and 402, a memory is connected to controller 122, preferably by inserting a MMC card 533 (FIG. 5) into memory connector 244. If the memory contains a firmware update, the controller recognizes this at 356 (FIG. 18). Controller 120 then queries itself and each slave device 201, 202 in the system and recognizes at 358 that the system does not contain a level of firmware equal to or later than that of the MMC card 533. It then checks 360 if the firmware in memory 533 is appropriate for the system. If these conditions are met, the firmware is uploaded to the appropriate memories in system 120. The program via process 364 checks each device in the entire system 120, and for each device performs the firmware update process for that device as appropriate.

Please replace the paragraph beginning on page 44, line 14, with the following:

Scene buttons screen objects 614 allow a combination of devices to be set at the same time, each to a different value. <u>Optionally, a SCENE CHANGE PASSWORD screen 616 may be programmed between SCENE button 614 and SCENE CHANGE screen 618.</u> The number of devices controlled is selectable. In the simplest case, a scene button screen objects may control only one setting of a single device, such as a button that, when pressed, turns a fireplace alternately on and off, or a button that sets one level to a lamp. A complex example is a button that might be labeled "FIRESIDE" which dims the main room lights, turns on mood lights over art objects, turns on the fireplace, and switches on the whole-house audio system to soft classical music. Scene button screen objects may be linked in groups that control the same devices but have independent levels, such as a set of lighting buttons labeled as "DINNER", "CLEAN", "CARDS", AND "OFF". In such a case, the DINNER button sets the lighting and music in the room, which is a dining room, at a relatively low level for eating, the CLEAN button sets the lighting at a full-on level and the music at an intermediate level for clean-up after eating, the CARDS button that sets the

lighting at an intermediate level and the music at a low level for games, such as cards, and the OFF button turns the lighting and music off. Display 250 in FIG. 10 shows MUSIC, READ, MOVE, CLEAN, and ALL OFF scene buttons.

Please replace the paragraph beginning on page 48, line 12, with the following:

By addition of an optional sprinkler control module, usually in cabinet 172, system 120 is able to control lawn sprinklers, water features, and other similar systems. Like other subsystems, this has the advantage that the sprinklers may be controlled from any room in the house. Pressing SPRINKLER screen object button [[660]] 661 brings up a SPRINKLER CONTROL screen, which shows the status of these systems and permits testing and other manual control. Further screens 666 allow the schedule of events of these devices to be programmed. If the weather subsystem is also installed, sprinkling may be controlled according to the measured rainfall, and features may be disabled while rain is occurring. The sprinkler system is preferably linked to the main time reference, so no resetting is needed for daylight savings time transitions. An optional SPRINKLER ACCESS PASSWORD screen 662 may be inserted between object [[660]] 661 and screen 664, and an optional SPRINKLER SCHEDULE PASSWORD screen 665 may be inserted between screens 664 and 666. The sprinkler programming is iterative along path 668 until all sprinkler functions are set.

Please replace the paragraph beginning on page 48, line 27, with the following:

As also discussed above in connection with FIGS. 11 and 15, PROGRAM screen object button 1110 brings up a PROGRAM DEVICES SELECT screen 195 (FIG. 15) which lists all the programmable devices in system 120 and shows the room in which each device is located. Underlying list screen 195 is a group of screens 675, which permit the programming of the screens for each interface unit 202 in the building. PROGRAM button 1110 may optionally require entry of a security code at screen 671 to permit programming access. If the device selected to be programmed is a touchscreen user interface unit, screen 195 permits one to go to either a DEVICE ATTRIBUTE SELECT screen 674 or a

SCREEN OBJECT SELECT screen 676. If the device is not a touchscreen user interface unit, the only choice is the DEVICE ATTRIBUTE SELECT screen 674, which is displayed on the default programming screen. When first programming a touchscreen, it is necessary to go to SCREEN OBJECT SELECT screen 676. This screen permits one to select and add, move, or remove screen objects. Screen objects are first selected via SCREEN OBJECT ADD SELECT screen 680, from a list of available objects. The selected objects are then given attributes via SCREEN OBJECT ATTRIBUTE SELECT screen 678. Examples of attributes that may be selected for a screen object are room name, label, such as "west wall", minimum value, maximum value, fade time, auto-on time, auto-off time, lighting curve, and linked devices. "Fade time" is a term for how long it takes a level control to go from the maximum to minimum setting and vice-versa. The lighting curve describes the correspondence between the setting of a level control and the power supplied. The linked device attribute describes how many devices are to be linked to a screen object. If the attribute entry is something that requires alphanumeric entries, such as the room name, the system goes to ALPHANUMERIC ATTRIBUTE EDIT ABC screen 696, which allows entry of letters of the alphabet. The programmer can switch between screen 696 and an ALPHANUMERIC ATTRIBUTE EDIT 123 screen 698, which permits the entry of numbers. If the attribute entry is one that requires only numbers, such as a maximum value, the system goes to NUMERIC ATTRIBUTE EDIT screen 684. If the attribute requires a choice of several alternatives, the system goes to MULTIPLE CHOICE ATTRIBUTE EDIT screen 686. If an attribute is a device linkage, the system goes to a DEVICE LINKAGE EDIT screen 688 from which a linkage can be added at 692 or deleted at 694. If the attribute requires time/date information, the system goes to a TIME/DATE EDIT screen 690. Once a screen object is selected, it can be removed via a SCREEN OBJECT DELETE button, which brings up a SCREEN OBJECT REMOVE WARNING screen 682, which allows the programmer to delete the object. Screen objects selected are listed on SCREEN OBJECT SELECT screen 676. A scroll button on this screen allows the programmer to select a screen object. Screen objects can be moved simply by selecting them on screen 676 and moving them to another line of the listing. If the device

to be programmed is not a touchscreen, or if it is desired only to program the attributes of the touchscreen or screen objects associated with the touchscreen, the program proceeds to DEVICE ATTRIBUTE SELECT screen 674. Depending on the attribute selected, the system displays screens 696, 684, 686, or 688, which are used as discussed above.